



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/709,974	06/10/2004	Heping Shang	SYB/0105.00	3973
31779	7590	08/17/2006	EXAMINER	
JOHN A. SMART 708 BLOSSOM HILL RD., #201 LOS GATOS, CA 95032-3503			PULLIAM, CHRISTYANN R	
			ART UNIT	PAPER NUMBER
			2191	

DATE MAILED: 08/17/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

## Office Action Summary

Application No.

10/709,974

Applicant(s)

SHANG ET AL.

Examiner

Christyann Pulliam

Art Unit

2191

— The MAILING DATE of this communication appears on the cover sheet with the correspondence address —

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 6/17/2004 and 4/20/2005.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-44 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-44 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 17 June 2004 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)                        | 4) <input type="checkbox"/> Interview Summary (PTO-413)                     |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)               | Paper No(s)/Mail Date. _____  |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date <u>8/31/2004</u> .   | 6) <input type="checkbox"/> Other: _____                                    |

### **DETAILED ACTION**

1. Claims 1-44 are pending as filed June 10, 2004.

#### ***Specification***

2. The abstract of the disclosure is does not commence on a separate sheet in accordance with 37 CFR 1.52(b)(4). A new abstract of the disclosure is required and must be presented on a separate sheet, apart from any other text. The title cannot be on the same page as the abstract. Appropriate correction required.

3. Applicant is reminded of the proper language and format for an abstract of the disclosure. The language should be clear and concise and should not repeat information given in the title. It should avoid using phrases which can be implied, such as, "The disclosure concerns," "The disclosure defined by this invention," "The disclosure describes," etc. The phrase "is described" appears twice in the abstract. Appropriate correction is required.

4. The disclosure is objected to because of the following informalities: spelling error. "Alist" in paragraph [0063] of the specification as filed. Appropriate correction is required.

5. The disclosure is objected to because of the following informalities: code segments should be labeled and those labels should be used in the citation of code lines. Appropriate correction is required.

### ***Claim Objections***

6. The claims are objected to because of the following informalities: the lack of the phrase "We claim" at the start of the claims. See MPEP § 608.01(m). Appropriate correction is required.

7. Claim 41 is objected to because of the following informalities: the claim is out of order. A claim that depends from a dependent claim should not be separated by any claim that does not also depend from said dependent claim. See MPEP § 608.01(n). For example, since Claim 41 depends on Claim 38 and it is the only claim that depends from Claim 38, it should directly follow Claim 38. Appropriate correction is required.

### ***Claim Rejections - 35 USC § 101***

8. 35 U.S.C. § 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

9. Claims 1-22 and 44 are rejected under 35 U.S.C. § 101 because the claims are directed toward non-statutory material.

10. Claims 1-22 and 44 are rejected under 35 U.S.C. § 101 because the claims are functional descriptive material. The system in Claim 1 is software per se, which is functional descriptive material and therefore non-statutory. Claim 1 lacks the necessary physical structure to be a machine or article of manufacture. Claims 2-22 depend from

Claim 1 and do not add any physical structure. Therefore, Claims 2-22 are also functional descriptive material. Additionally, Claim 44 is software per se, which is functional descriptive matter. It lacks the necessary physical structure to make it an article of manufacture or a machine. Claims 1-22 and 44 are directed to non-statutory material.

11. Claims 1-22 are rejected under 35 U.S.C. § 101 because the claims lack a tangible result. The final action in Claim 1 is "determining subscribers". This "determining" is not a tangible result. Claims 2-22 do not add a tangible result to the system of Claim 1. There is no tangible result that would make the determination useful. Therefore, Claims 1-22 are non-statutory.

***Claim Rejections - 35 USC § 102***

12. The following is a quotation of the appropriate paragraphs of 35 U.S.C. § 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

13. Claims 1-8, 22-28 and 42-44 are rejected under 35 U.S.C. § 102(b) as being anticipated by Zarmer et al., U.S. Patent No. 5,625,818 (hereinafter Zarmer).

14. As for Claim 1, Zarmer teaches:

A system for determining subscribers to which a published item of data should be replicated, the system comprising:

an interface module (See e.g. col. 5, lines 2-10 and col. 7, lines 5-10) for receiving user input of lists of a plurality of subscribers, each list specifying items of data to be replicated to a given subscriber (See e.g. subscribing col. 13, lines 2-10);

a build module for building an index based on the lists of the plurality of subscribers containing entries indicating subscribers for each item of data specified in the lists (See e.g. ObjectMan can search list so it was built – col. 11, lines 12-38) and a default list of subscribers for items of data not matching any of the entries (See e.g. auto-interest col. 22, lines 42-53); and

a resolution module for receiving a published item of data and determining subscribers to which the published item should be replicated based on the index (See e.g. ObjectMan – col. 11, lines 12-38).

15. As for Claim 2, Zarmer also teaches:

The system of claim 1, wherein a subscriber comprises a replicate database (See e.g. IP is subscriber database col. 9, line 17-60 and col. 16, lines 52-64 and Claim 1 – second database portion is a substantial replica of the first).

16. As or Claim 3, Zarmer also teaches:

The system of claim 1, wherein a published item of data comprises an item of data published by a primary database for replication (See e.g. CS is publisher database col. 9, line 17-60, col. 7, lines 5-10 and col. 8, lines 32-40 and Claims 1 and 2).

17. As for Claim 4, Zarmer also teaches:

The system of claim 1, wherein the lists of the plurality of subscribers include a list specifying at least one item of data not to be replicated to a particular subscriber

(See e.g. RemoveInterestedView – col. 14, lines 22-27 and RemoveInterestedParties col. 28, line 65- col. 29, line 35).

18. As for Claim 5, Zarmer also teaches:

The system of claim 1, wherein the lists of the plurality of subscribers include a list comprising a negation set indicating that all data other than items specified on the list should be replicated to a particular subscriber (See e.g. col. 4, line 66- col.5, line 7 and 26-45, and RemoveInterestedView – col. 14, lines 22-27 and RemoveInterestedParties col. 28, line 65- col. 29, line 35).

19. As for Claim 6, Zarmer also teaches:

The system of claim 1, wherein the interface module receives at least one string identifying data to be replicated to a subscriber (See e.g. AddInterestedView col. 14, lines 20-23 and col. 13, lines 3-8 and AddInterestedParties col. 28, line 65- col. 29, line 35).).

20. As for Claim 7, Zarmer also teaches:

The system of claim 6, wherein the build module builds the index based, at least in part, on said at least one string (See e.g. ObjectMan – col. 11, lines 12-38).

21. As for Claim 8, Zarmer also teaches:

The system of claim 7, wherein the index includes an entry representing a particular item of data and at least one subscriber to which the particular item of data is to be replicated (See e.g. ObjectMan – col. 11, lines 12-38).

22. As for Claim 22, Zarmer also teaches:

The system of claim 1, wherein the build module removes a subscriber from the index in response to a request to remove a subscriber (See e.g. RemoveInterestedView – col. 14, lines 22-27).

23. As for Claims 23, 43 and 44, Zarmer teaches:

A method for determining subscribers to which a published item of data should be replicated, the method comprising:

receiving lists of a plurality of subscribers, each list specifying items of data to be replicated to a given subscriber (See e.g. subscribing col. 13, lines 2-10);

building an index based on said lists of the plurality of subscribers containing entries indicating subscribers to which each item of data specified in said lists should be replicated (See e.g. ObjectMan can search list so it was built – col. 11, lines 12-38) and a default list for indicating subscribers to published items of data not matching any of the index entries (See e.g. auto-interest col. 22, lines 42-53);

given a published item of data, determining whether the published item matches at least one entry in the index (See e.g. ObjectMan can search list so it was built – col. 11, lines 12-38);

if the published item matches at least one entry, generating a set of subscribers to which the published item should be replicated based on the subscribers associated with said at least one entry (See e.g. ObjectMan can search list so it was built – col. 11, lines 12-38 and AutoBroadcast col. 15, lines 24-30); and



otherwise, returning the default list of subscribers to which the published item should be replicated (See e.g. col. 22, lines 43-50).

24. As for Claim 24, Zarmer also teaches:

The method of claim 23, wherein a subscriber comprises a replicate database (See e.g. IP is subscriber database col. 9, line 17-60 and col. 16, lines 52-64 64 and Claim 1 – second database portion is a substantial replica of the first).

25. As for Claim 25, Zarmer also teaches:

The method of claim 23, wherein a published item of data comprises an item of data published by a primary database for replication (See e.g. CS is publisher database col. 9, line 17-60, col. 7, lines 5-10 and col. 8, lines 32-40 and Claims 1 and 2).

26. As for Claim 26, Zarmer also teaches:

The method of claim 23, wherein said receiving step includes specifying items of data not to be replicated to a subscriber (See e.g. RemoveInterestedView – col. 14, lines 22-27 and RemoveInterestedParties col. 28, line 65- col. 29, line 35).

27. As for Claim 27, Zarmer also teaches:

The method of claim 23, wherein said receiving step includes receiving a list comprising a negation set indicating that all data other than items specified on the list should be replicated to a subscriber (See e.g. col. 4, line 66- col.5, line 7 and 26-45, and RemoveInterestedView – col. 14, lines 22-27 and RemoveInterestedParties col. 28, line 65- col. 29, line 35).

28. As for Claim 28, Zarmer also teaches:

The method of claim 23, wherein said receiving step includes receiving at least one string identifying data to be replicated to a subscriber (See e.g. ObjectMan – col. 11, lines 12-38).

29. As for Claim 42, Zarmer also teaches the determining step includes building a list of subscribers based upon a plurality of matching entries (See e.g. col. 11, lines 30-38).

***Claim Rejections - 35 USC § 103***

30. The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

31. Claims 9-11, 13-19, 21, 29-37 and 39-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zarmer et al., U.S. Patent No. 5,625,818 (hereinafter Zarmer) as applied above, and in view of Pedrizetti et al., U.S. Patent No. 6,151,708 (hereinafter Pedrizetti).

32. As for Claim 9, Zarmer teaches the system of claim 1. Zarmer does not teach the use of a hash table. However, Pedrizetti teaches a hash table of entries based on the lists of the plurality of subscribers (See e.g. Claim 2 and col. 4, lines 55-58).

33. Zarmer and Pedrizetti are from the analogous art of updating remote computers and databases from a main server. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined Zarmer and Pedrizetti.

34. The motivation to combine Zarmer and Pedrizetti comes from the common goal of efficient automation. Zarmer discusses the problems with manually tracking published data (See col. 1, lines 30-55). Pedrizetti discusses the issues with large updates and version control (See col. 1, lines 12-40). Both try to overcome these issues by automating the version control and data transfers over networks to interested computers. Pedrizetti adds the well-known hash table to the file transfer system of Zarmer in order to determine which data goes to which interest client computer. Zarmer and Pedrizetti allow a server computer to publish data to client computers that have expressed an interest in the data in an efficient automated process.

35. As for Claim 10, Zarmer teaches the system of Claim 1. Zarmer does not teach a bitmap string. However, Pedrizetti teaches a bitmap string representing subscribers to which an item of data is to be replicated (See e.g. col. 4, lines 1-13 and 50-54).

36. As for Claim 11, Zarmer teaches the system of Claim 1. Zarmer does not teach a bitmap string. However, Pedrizetti teaches the bitmap string includes one bit for each subscriber (See e.g. col. 5, lines 26-45 and col. 4, lines 50-54).

37. As for Claim 13, Zarmer teaches the system of Claim 1. Zarmer does not teach a bitmap string. However, Pedrizetti teaches a bitmap string representing the default list of subscribers (See e.g. client is subscriber - col. 3, lines 40-57 and col. 4, line 66- col.5, line 7).

38. As for Claim 14, Zarmer as modified above teaches the system of Claims 1 and 13. Zarmer does not teach adding subscribers to the list. However, Pedrizetti teaches adding each subscriber having a list comprising a whole set to the bitmap string representing the default list of subscribers (See e.g. col. 4, line 66- col.5, line 7 and 26-45).

39. As for Claim 15, Zarmer as modified above teaches the system of Claims 1 and 13. Zarmer does not teach adding subscribers to the list. However, Pedrizetti teaches adding each subscriber having a list comprising a negation set to the bitmap string representing the default list of subscribers (See e.g. col. 4, line 66- col.5, line 7 and 26-45).

40. As for Claim 16, Zarmer teaches the system of Claim 1. Zarmer does not teach a hash value for items to be published. However, Pedrizetti teaches a hashed value based on at least one string identifying the published item (See e.g. col. 4, lines 1-17, 55-59).

41. As for Claim 17, Zarmer as modified above teaches the system of Claims 1 and 16. Zarmer does not teach a hash value for items to be published. However, Pedrizetti teaches searches for the hashed value in the index (See e.g. col. 5, lines 7-45).

42. As for Claim 18, Zarmer as modified above teaches the system of Claims 1 and 17. Zarmer does not teach a hash value for items to be published. However, Pedrizetti teaches determines subscribers to which the published item should be replicated based upon at least one entry in the index including the hashed value if the hashed value is found in the index (See e.g. col. 5, lines 7-45 and col. 6, lines 45-48).

43. As for Claim 19, Zarmer as modified above teaches the system of Claims 1 and 17. Zarmer does not teach a hash value for items to be published. However, Pedrizetti teaches determining subscribers to which the published item should be replicated based upon the default list if the hashed value is not found in the index (See e.g. col. 3, lines 5-41).

44. As for Claim 21, Zarmer teaches the system of Claim 1. Zarmer does not teach a hash value for items to be published. However, Pedrizetti teaches determining subscribers to which the published item should be replicated based upon the default list of subscribers for published items without matching entries in the index (See e.g. col. 3, lines 5-41).

45. As for Claim 29, Zarmer teaches the system of Claims 23 and 28. Zarmer does not teach a hash table. However, Pedrizetti teaches building a hash table based on said at least one string (See e.g. Claim 2 and col. 4, lines 55-58).

46. As for Claim 30, Zarmer as modified above teaches the system of Claims 23, 28 and 29. Zarmer does not teach a bitmap string. However, Pedrizetti teaches building a bitmap string for each entry representing subscribers to an item of data (See e.g. col. 4, lines 1-13 and 50-54).

47. As for Claim 31, Zarmer teaches the system of Claim 23. Zarmer does not teach a hash table. However, Pedrizetti teaches building a hash table of entries (See e.g. Claim 2 and col. 4, lines 55-58).

48. As for Claim 32, Zarmer teaches the system of Claim 23. Zarmer does not teach a bitmap string. However, Pedrizetti teaches generating a bitmap string representing subscribers to an item of data (See e.g. col. 4, lines 1-13 and 50-54).

49. As for Claim 33, Zarmer as modified above teaches the system of Claims 23 and 32. Zarmer does not teach a bitmap string. However, Pedrizetti teaches generating a bitmap string having one bit for each subscriber (See e.g. col. 5, lines 26-45 and col. 4, lines 50-54).

50. As for Claim 34, Zarmer as modified above teaches the system of Claims 23 and 32. Zarmer also teaches wherein said step of generating a bitmap string includes removing a subscriber from the bitmap string representing subscribers to an item of data if the subscriber has specified the item of data as an excluded item in a negation set (See e.g. col. 4, line 66- col.5, line 7 and 26-45, and RemoveInterestedView – col. 14, lines 22-27 and RemoveInterestedParties col. 28, line 65- col. 29, line 35).

51. As for Claim 35, Zarmer teaches the system of Claim 23. Zarmer does not teach a bitmap string. However, Pedrizetti teaches building a bitmap string for representing the default list of subscribers (See e.g. client is subscriber - col. 3, lines 40-57 and col. 4, line 66- col.5, line 7).

52. As for Claim 36, Zarmer as modified above teaches the system of Claims 23 and 35. Zarmer does not teach a bitmap string. However, Pedrizetti teaches the step of building a bitmap string for representing the default list of subscribers includes adding each subscriber having a list comprising a whole set (See e.g. col. 4, line 66- col.5, line 7 and 26-45).

53. As for Claim 37, Zarmer as modified above teaches the system of Claims 23 and 35. Zarmer does not teach a bitmap string. However, Pedrizetti teaches the step of building a bitmap string for representing the default list of subscribers includes adding each subscriber having a list comprising a negation set (See e.g. col. 4, line 66- col.5, line 7 and 26-45).

54. As for Claim 39, Zarmer teaches the system of Claim 23. Zarmer does not teach a hashed value. However, Pedrizetti teaches the determining step includes generating a hashed value based on at least one string identifying the published item (See e.g. col. 4, lines 1-17, 55-59).

55. As for Claim 40, Zarmer as modified above teaches the system of Claims 23 and 39. Zarmer does not teach a hashed value. However, Pedrizetti teaches the determining step includes searching for the hashed value in the index (See e.g. col. 5, lines 7-45).

56. Zarmer and Pedrizetti are from the analogous art of updating remote computers and databases from a main server. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined Zarmer and Pedrizetti.

57. The motivation to combine Zarmer and Pedrizetti comes from the common goal of efficient automation. Zarmer discusses the problems with manually tracking published data (See col. 1, lines 30-55). Pedrizetti discusses the issues with large updates and version control (See col. 1, lines 12-40). Both try to overcome these issues by automating the version control and data transfers over networks to interested computers. Pedrizetti adds the well-known hash table to the file transfer system of

Zarmer in order to determine which data goes to which interest client computer. Zarmer and Pedrizetti allow a server computer to publish data to client computers that have expressed an interest in the data in an efficient automated process.

58. Claims 12 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zarmer et al., U.S. Patent No. 5,625,818 (hereinafter Zarmer) as applied above, and in view of Collison, U.S. PGPub No. 2004/0139166 (hereinafter Collison).

59. As for Claim 12, Zarmer teaches the system of Claim 1. Zarmer does not teach wildcards. However, Collison teaches an entry including a wildcard for indicating all items of data of a certain type should be replicated to subscribers of said entry (See e.g. paragraphs [0026-0031]).

60. Zarmer and Collison are from the analogous art of communicating with remote computers and databases from a main server. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined Zarmer and Collison.

61. The motivation to combine Zarmer and Collison comes from the common goal of efficient communication of data to multiple computers over a network. Zarmer and Collison allow computers to subscribe to receive data or messages that they have an interest in receiving. Collison adds flexibility to the Zarmer system by allowing the use of wildcards in the subscribing process. Wildcards are a common tool to allow clients to find things without being precise in their selections. Since the subscribers to the Zarmer system would naturally want to subscribe to all topics in their interest areas not just the



ones they knew the exact names of, it would be logical to use the common tool of wildcards as presented in Collison.

62. As for Claim 38, Zarmer teaches the system of Claim 23. Zarmer does not teach wildcards. However, Collison teaches building entries including a wildcard for indicating all items of data of a certain type should be replicated to a subscriber (See e.g. paragraphs [0026-0031].

63. Zarmer and Collison are from the analogous art of communicating with remote computers and databases from a main server. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined Zarmer and Collison.

64. The motivation to combine Zarmer and Collison comes from the common goal of efficient communication of data to multiple computers over a network. Zarmer and Collison allow computers to subscribe to receive data or messages that they have an interest in receiving. Collison adds flexibility to the Zarmer system by allowing the use of wildcards in the subscribing process. Wildcards are a common tool to allow clients to find things without being precise in their selections. Since the subscribers to the Zarmer system would naturally want to subscribe to all topics in their interest areas not just the ones they knew the exact names of, it would be logical to use the common tool of wildcards as presented in Collison.

65. Claims 20 and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zarmer et al., U.S. Patent No. 5,625,818 (hereinafter Zarmer) in view of Pedrizetti

et al., U.S. Patent No. 6,151,708 (hereinafter Pedrizetti) as applied above, and further in view of Collison, U.S. PGPub No. 2004/0139166 (hereinafter Collison).

66. As for Claim 20, Zarmer as modified above by Pedrizetti teaches the system of Claims 1 and 16. Zarmer does not teach wildcards. However, Collison teaches a plurality of strings identifying the published item and substitutes a wildcard for one of the strings to search for matching entries in the index which include a wildcard (See e.g. paragraphs [0026-0031]).

67. Zarmer and Pedrizetti are from the analogous art of updating remote computers and databases from a main server. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined Zarmer and Pedrizetti.

68. Zarmer and Collison are from the analogous art of communicating with remote computers and databases from a main server. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined Zarmer and Collison.

69. The motivation to combine Zarmer and Pedrizetti comes from the common goal of efficient automation. Zarmer discusses the problems with manually tracking published data (See col. 1, lines 30-55). Pedrizetti discusses the issues with large updates and version control (See col. 1, lines 12-40). Both try to overcome these issues by automating the version control and data transfers over networks to interested computers. Pedrizetti adds the well-known hash table to the file transfer system of Zarmer in order to determine which data goes to which interest client computer. Zarmer

and Pedrizetti allow a server computer to publish data to client computers that have expressed an interest in the data in an efficient automated process.

70. The motivation to combine Zarmer and Collison comes from the common goal of efficient communication of data to multiple computers over a network. Zarmer and Collison allow computers to subscribe to receive data or messages that they have an interest in receiving. Collison adds flexibility to the Zarmer system by allowing the use of wildcards in the subscribing process. Wildcards are a common tool to allow clients to find things without being precise in their selections. Since the subscribers to the Zarmer system would naturally want to subscribe to all topics in their interest areas not just the ones they knew the exact names of, it would be logical to use the common tool of wildcards as presented in Collison.

71. As for Claim 41, Zarmer as modified above by Pedrizetti teaches the system of 23 and 38. Zarmer does not teach wildcards. However, Collison teaches receiving a plurality of strings identifying the published item and substituting a wildcard for one of the strings to search for matching entries in the index which include a wildcard (See e.g. paragraphs [0026-0031]).

72. Zarmer and Pedrizetti are from the analogous art of updating remote computers and databases from a main server. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined Zarmer and Pedrizetti.

73. Zarmer and Collison are from the analogous art of communicating with remote computers and databases from a main server. It would have been obvious to one of

ordinary skill in the art at the time the invention was made to have combined Zarmer and Collison.

74. The motivation to combine Zarmer and Pedrizetti comes from the common goal of efficient automation. Zarmer discusses the problems with manually tracking published data (See col. 1, lines 30-55). Pedrizetti discusses the issues with large updates and version control (See col. 1, lines 12-40). Both try to overcome these issues by automating the version control and data transfers over networks to interested computers. Pedrizetti adds the well-known hash table to the file transfer system of Zarmer in order to determine which data goes to which interest client computer. Zarmer and Pedrizetti allow a server computer to publish data to client computers that have expressed an interest in the data in an efficient automated process.

75. The motivation to combine Zarmer and Collison comes from the common goal of efficient communication of data to multiple computers over a network. Zarmer and Collison allow computers to subscribe to receive data or messages that they have an interest in receiving. Collison adds flexibility to the Zarmer system by allowing the use of wildcards in the subscribing process. Wildcards are a common tool to allow clients to find things without being precise in their selections. Since the subscribers to the Zarmer system would naturally want to subscribe to all topics in their interest areas not just the ones they knew the exact names of, it would be logical to use the common tool of wildcards as presented in Collison.

**Conclusion**

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Christyann Pulliam whose telephone number is 571-270-1007. The examiner can normally be reached on M-Th 7:30 am-5 pm, every other Fri 7:30am-4pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David Bruce can be reached on 571-272-2487. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

CRFP

  
DAVID BRUCE  
SUPERVISOR, PATENT EXAMINER